An Assessment of the Impact of Decommissioning the Cray SV1 Systems at NERSC

Jonathan T. Carter

National Energy Research Scientific Computing Center

Ernest Orlando Lawrence Berkeley National Laboratory
Berkeley, CA 94720

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Sciences Division, U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Introduction

The line of Cray Parallel Vector Processors (PVP) running the Unicos operating system is coming to an end with the SV1. Cray's successor system, the SV2 due out in the last half of 2002, will be based more on IRIX and Unicos/mk technology than on Unicos and will be considerably different than the current SV1 systems. NERSC users have had continuous access to Cray hardware since 1981, and Cray Unicos systems since 1992. The transition from a traditional Cray platform is inevitable, so a review of the current usage of the systems was undertaken in order to better understand the major issues that are likely to occur during the migration.

In this report, we will make frequent comparison to the NERSC IBM SP system. This system is typical (aside from its size) of commercial SMP systems currently available. Even though the replacement for the Cray SV1 cluster is presently undecided, our experience with the SP enables concrete comparisons to be made between important features of the Cray SV1 and similar features of another architecture.

User Survey

A survey of the major SV1 users from each office was conducted in the first few months of 2001. The users were asked the major reasons for their use of the SV1 systems, and whether they had identified any barriers to moving away from the SV1 systems.

The responses, summarized by Program or Office, are shown below.

Biological and Environmental Research, Environmental Science

- Mathematical libraries NAG and IMSL are important for program development.
- Some programs have been tuned for vector architecture, and some are enabled for shared-memory parallelism.
- Good interactive computing resources are required.

Biological and Environmental Research, Life Sciences and Medical Sciences

- Chemistry application packages (Gaussian 98 and GAMESS) are used extensively.

Basic Energy Science, Chemical Science

- Chemistry application packages (Gaussian 98, MOLPRO and GAMESS) are used extensively.

- Some packages have large memory requirements, up to 4GB, and large disk space requirements, up to 40GB.

Basic Energy Science, Materials Science

- The majority of users funded by this office predicted they would move away from the SV1 platform.
- Older applications were developed for vector architecture, but new designs were more likely to be distributed-memory MPI programs.
- Scientific and mathematical libraries are important for program development, as are simple visualization tools (NCAR library).
- The length of batch queue time limits is important, as breaking up jobs into very many steps is tedious.

Fusion Energy Science

- Most users from this office are in the process of porting programs to a distributed-memory MPI paradigm, but still see the need for large-scale shared-memory platform. This is because many older programs are not easily modified to run on distributed-memory machines, and the development of new prototype applications is easier on a shared-memory machine.
- The hardware itself is not as important a factor as the environment is for developing programs. This includes access to mathematical and graphics libraries, and also use of NERSC User Services Group consulting help.
- A major issue is the length of the batch queue time limits, with good interactive access required.
- In addition, many users noted the need for CTSS to Unix text file conversion utilities, and a way to read Cray unformatted files on non-Cray platforms.

High Energy and Nuclear Physics

- Codes are often originally written for vector architecture and shared-memory parallelism, but in many cases also run on distributed-memory machines.
- Large memory is often required, up to 4GB.
- Mathematical libraries are used fairly extensively.
- The flexibility, and extensions provided by the Cray f90 compiler are important to keep “dusty decks” running.

Hardware Considerations

This section will attempt to identify possible stumbling blocks in the migration away from the current SV1 hardware. As mentioned in the Introduction, this report will try to quantify the situation by comparing the current Cray hardware primarily with the IBM SP system, and with other newer hardware in the marketplace. The areas of “real”

application performance and shared-memory parallelism were often mentioned in the survey and treated separately below.

In terms of CPU performance, the Cray SV1 systems have a rather modest clock speed of 300 MHz compared with many RISC processors. The vector architecture and cache hide memory latency very effectively, giving sustained memory bandwidths greater than the fastest current RISC SMP nodes. Clearly, the extent of vectorization and memory access pattern strongly influences real performance. See the Application Performance section below for some actual studies.

Each Cray SV1 chassis has 8 GB of memory per 24 processors. By current SMP standards this is relatively small, the NERSC IBM SP having typically 16 GB per 16 processors, and as much as 64 GB per 16 processors on some nodes. Each SV1 has around 330 GB of disk space, with batch job limits of up to 40 GB per job. Again, this is relatively modest by large SMP cluster standards.

Application Performance

In this section we present a series of benchmarks comparing the performance of synthetic benchmarks, application codes, and user codes. Results for the NAS Parallel Benchmarks 2.3 (serial version) are shown below. These benchmarks comprise several kernels and small applications designed to mimic the behavior of computational fluid dynamics programs.

Benchmark	*Cray SV1 (MFlop/s)*	*IBM PWR3+ (MFlop/s)*	*Ratio*
BT	64	143	2.23
CG	68	97	1.43
FT	125	145	1.16
LU	94	285	3.03
MG	161	202	1.25
SP	117	106	0.91

Survey results from both the Chemical Science and Life Science programs show extensive use of commercial or community application programs. For this reason the performance of the **Gaussian 98** and **MOLPRO** packages were benchmarked on both the Cray SV1 and on a single CPU of the NERSC IBM SP. A selection of different calculations was performed with each code, some methods being CPU intensive, and some I/O intensive. The types of calculations are considered representative of the kinds of calculations performed by NERSC users. Because of differences in method and simulation inputs, the results are not comparable between the two packages, only between the two platforms.

Benchmark	*Cray SV1 (secs.)*	*IBM PWR3+ (secs.)*	*Ratio*
	Gaussian 98		
DFT/Freq	7527	2306	3.26
MP2/Energy	3024	1766	1.71
CCSD(T)/Energy	10070	9312	1.08
	MOLPRO		
DFT/Force	6580	2263	2.91
MP2/Energy	4042	965	4.19
CCSD(T)/Energy	5909	3263	1.81

Lastly, a user application code was taken from the Cray platform, ported to the IBM, and benchmarked. The code, **tsc**, from Princeton Plasma Physics Laboratory is well vectorized, running at around 100 MFlop/s on the Cray SV1.

Benchmark	*Cray SV1 (secs.)*	*IBM PWR3+ (secs.)*	*Ratio*
tsc	905	551	1.64

From these results, it seems that a modern RISC-based SMP architecture can deliver performance in excess of the traditional SV1 platform.

Shared-memory Parallelism

The Cray SV1 supports shared-memory parallelism through Cray Autotasking directives and OpenMP directives. However, Cray Autotasking directives have been deprecated for some time now and most users are developing programs using OpenMP.

Most modern SMP systems support shared-memory parallelism via OpenMP through vendor compilers, or products such as the KAP/Pro Toolset from Intel. The KAP/Pro Toolset can convert simple Cray Autotasking directives into corresponding OpenMP directives.

Software Considerations

It is apparent from the survey results that mathematical libraries and third-party vendor applications are very important to a large number of SV1 users. In addition, tools to convert Cray or CTSS data files to future platforms were requested by multiple users. Lastly, the various language extensions and Cray library calls that have become incorporated in user's applications will have to be substituted, or equivalents rewritten.

Scientific, Mathematical, and I/O Libraries

The following table shows the status of the libraries currently installed on the SV1 cluster. The majority of these libraries are available on other platforms, and many have been installed on the IBM SP.

Library	Description	Status	Installed on IBM SP
harwell	General Mathematical Library	Updated version, Most platforms, Better alternative	No
hdf	Portable-format I/O library	Most platforms	Yes
imsl	General Mathematical Library	Most platforms	Yes
lsode	Ordinary Differential Equations Solver	Cray source available	No
nag	General Mathematical Library	Most platforms	Yes
ncar	Graphics Software	Most platforms	Yes
netCDF	Portable-format I/O library	Most platforms	Yes
pact	Portable Application Code Toolkit	Most platforms	No
slatec	General Mathematical Library	Source available, Better alternative	No
libcbase	General Utility Routines	Cray source available	No
libstack	General Utility Routines	Cray source available	No
libtv80	Graphics Software	Source available, Better alternative	No
libgraf	Graphics Software (graflib)	Cray source available	No
libcore	Graphics Software (graflib)	Cray source available	No
libxtcmds	Graphics Software (graflib)	Cray source available	No
basis	Programming Framework	Updated version, Most platforms	No

In the Status column, “Most platforms” indicates there is some external support for the product on most platforms; “Source available” indicates there is no external support, and the source code is available as-is; “Cray source available” indicates that NERSC has source code, but it contains dependencies on the Cray architecture (possibly including assembly language); “Updated version” indicates that a more recent version than that installed on the SV1 systems is available with some external support. “Better Alternative” indicates the library is no longer being developed or supported, and the functionality exists in better-supported alternatives.

It is not anticipated that software libraries in the "Cray source available" category will be ported to other platforms because of the high software engineering costs. Users with dependencies on these libraries will be migrated to more portable equivalents. Software libraries in the category "Updated version" might not be entirely compatible with current use on the SV1 cluster, so some small alterations to application programs might be required.

Scientific Applications

The following table shows the status of all applications software on the Cray SV1 cluster.

Application	Description	Status	Installed on IBM SP
Amber	Molecular Dynamics	Most platforms	Parallel
ANSYS	Structural Analysis	Retired	No
CCM3	Climate Modeling	Most platforms	No
CSM	Climate Modeling	Updated version, Most platforms (2001)	No
Gaussian 98	Molecular Electronic Structure	Most platforms	Parallel
GAMESS	Molecular Electronic Structure	Most platforms	No
ITS	Nuclear Transport	Cray source available	No
MAFIA	Electromagnetic Fields Simulation	Cray source available	No
MCNP	Nuclear Transport	Cray source available	No
MOLPRO	Molecular Electronic Structure	Most platforms	Serial
NASTRAN	Structural Analysis	Most platforms	No

In the Status column, "Most platforms" indicates there is some external support for the product on most platforms; "Cray source available" indicates that NERSC has source code, but it contains dependencies on the Cray architecture (possibly including assembly language); "Updated version" indicates that a more recent version than that installed on the SV1 systems is available with some external support.

The final column of the table shows whether a parallel or serial (single processor) version of the software is available for the SP.

The software identified as important by the Chemical Sciences and Life Sciences users is generally available on most HPC platforms, and the majority of those packages have been installed on the IBM SP. In addition, several other Chemistry packages, such as **NWChem** and **Q-Chem**, that are not available for Cray SV1, have also been installed on the IBM SP.

We do not anticipate migrating the packages ITS, MAFIA, and MCNP to any other platform since they show very little usage.

File Conversion Utilities

Several users mentioned the need to be able to convert CTSS files into Unix ASCII files. The CTSS file utility programs ctou and rlib are available for many platforms, and have been installed on the IBM SP. These utilities enable the conversion of CTSS text files and libraries of text files to regular Unix ASCII files.

The lack of tools for converting CTSS and Unicos unformatted data to other portable formats has yet to be completely addressed. Scientific Computing Division at the National Center for Atmospheric Research has developed software to convert many Cray file formats to current Unix standards, but these tools have not yet been evaluated at NERSC.

Cray Programming Environment

Many programming APIs have been added to the Unicos operating system over its lifetime, and some of this functionality has become accepted as a de facto standard for HPC programming. In this section we summarize the main libraries and recommend alternatives.

Mathematical and Scientific Routines

A large number of intrinsic functions were added to the standard Fortran 77 set. For example, Bessel function, error function, and gamma function evaluation are available as function calls from Fortran.

IBM provides many of these functions in XL Fortran product. Additional functions are available from the NAG or IMSL libraries.

Cray also provides:

- BLAS levels 1, 2, and 3
- Tuned versions of the generally available LAPACK and LINPACK packages
- Cray specific one, two, and three dimensional FFT, linear digital filter, and convolution routines
- a number of out-of-core linear algebra routines, handling matrixes which are too large to fit into memory
- sparse linear system solvers
- special linear systems solvers

Almost all vendors provide the BLAS routines and a subset of LAPACK. The rest of LAPACK and LINPACK is freely available if required.

IBM provides FFT and sparse linear systems solvers in the ESSL library. A group within IBM has been working to provide wrapper routines that allow the application programmer to keep the Cray interface intact. In the port of the **tsc** program from the Cray to the IBM SP, a version of a Cray specific matrix inversion routine was constructed easily from ESSL routines and a small amount of additional coding.

Datatype Conversion Routines

A large number of routines to convert between various (now mostly obsolete) machine formats are callable from Fortran. Probably the most significant are routines to convert Cray unformatted (floating-point, integer, logical and character) data to IEEE unformatted data, and the reverse conversions.

There are currently no data conversion routines provided by IBM.

I/O Routines

Cray provides at least four distinct families of routines, in addition to regular Fortran I/O statements, for performing I/O. These different families offer control over such parameters as:

- asynchronous and synchronous transfer
- word or record level addressability
- specifying the structure of the file on physical disk

Moreover, a layered I/O library - Flexible File I/O (FFIO) - sits between the Fortran I/O subsystem and the raw system-level interface. This layer is configurable at runtime and features many filters, buffering algorithms, and data conversion utilities.

Modern I/O configurations have largely obviated the need for complex user-configurable I/O routines. For special cases, POSIX standard I/O calls provide complete flexibility. The IBM XL Fortran compiler and runtime environment also provides control of asynchronous and synchronous transfers, and buffering.

Miscellaneous Routines

Cray provided many functions to increase the capabilities of the Fortran 77 language. Over years of development these functions have become incorporated in some application codes. For example,

- memory allocation routines (including debug versions)
- Fortran POSIX interface to system services
- functions to aid vectorization (for example, conditional vector merge functions and bit vector manipulation routines)
- timing and resource (for example, the amount of time remaining to a batch job) query routines

Some of this functionality is handled well by other vendors, some less so. Specifically:

- With the Fortran 90 standard, memory allocation in Fortran programs has become completely portable. IBM compilers provide debug memory management libraries for both Fortran and C.
- Unfortunately, the Fortran POSIX API is not well supported by other vendors. However, the C language API is always available.
- Vector functions are not required on non-vector architectures, although some vendors provide these functions for compatibility.
- By and large, the timing and resource query functions are not as convenient to use on non-Cray platforms. For example, to write a function to obtain the time remaining for a batch job running under the IBM Loadleveler batch system requires extensive C language programming. In this case, NERSC would provide a simple library to return useful resource data.

Job Management

In this section we summarized issues of job management that are particular to the SV1 cluster at NERSC. Many users commented on the necessity of providing adequate interactive access, of providing a platform for shared-memory parallelism, and of keeping batch queues time limits reasonably long.

The Cray SV1 cluster is scheduled as a traditional time-sharing supercomputer. One node of the cluster handles mainly interactive work, while the other two are dedicated to batch work. All the SV1 machines are frequently oversubscribed with respect to both CPU and, to a lesser extent, memory. The machines must swap processes in and out of memory and suspend and resume execution of processes repeatedly. While this leads to some overhead, it enables more work to run simultaneously, rather than waiting in a queue. This enables interactive, debug, and smaller jobs to have a relatively fast turnaround time, while enabling some jobs to run for a long time.

This is in contrast to how most massively parallel computers (MPP), including those composed of SMP nodes, are run. In the case of MPP systems, user applications tend to have exclusive use of nodes for a much longer period of time than a typical SV1 time slice. This is necessary since most MPP operating systems incur considerably more overhead in swapping a typical job that does the Cray SV1. This leads to an operational model where work waits in a queue, and then has a long period of uninterrupted running. To overcome the problem of bad turnaround for interactive and debug jobs, NERSC provides dedicated resources on the IBM SP for interactive and debug jobs.

An advantage of MPP composed of clusters of SMP nodes, as opposed to those made up of single processor nodes, is that single nodes can be used to develop and run shared-memory parallel programs.

The following table compares the resource limits for both the SV1 cluster and the IBM SP.

	Interactive		Batch	
	Time	Memory	Time	Memory
Cray SV1	10 hours CPU	640 MB	120 hours CPU	4 GB
IBM SP	1 hour CPU (serial) 30 mins. 128 procs.	2 GB per proc. 16 GB (serial)	24 hours 512 procs. 8 hours 2048 procs.	2 GB per proc. 64 GB (serial)

The IBM SP time limits is shorter than that of the SV1, but the overall compute power delivered to a parallel job are much greater. Of course, this depends on migration to a distributed-memory, message-passing method of programming. However, even using the 16 processors of one node via OpenMP leads to an increase in resources over the SV1.

Summary

In this section we summarize the findings of the previous sections, and list any requirements for a successful migration from the SV1 cluster not addressed.

- Given the performance comparisons in the Hardware section of this report, it is evident that a workstation should be an adequate replacement for some of the serial applications running on the SV1 cluster. For example, many Intel based workstations now have better performance than an IBM Power3+ node on a per CPU basis, and the Power3+ showed a significant performance advantage over the SV1 in most of the benchmarks run in this report.
- For the large-scale SV1 applications, single nodes of the IBM SP should provide substantially more compute power. Porting programs to a distributed-memory MPI programming paradigm will enable users to take advantage of even greater compute resources.
- Job resource and time limits on the IBM SP provide access to equivalent, or larger, amounts of compute power than on the SV1 cluster. Interactive work is supported.
- For the important applications and libraries on the SV1, continuing support should be made available.
- Utilities should be provided to convert between Cray formatted and unformatted files, and CTSS formatted files, to current equivalents.
- Replacements for common Cray routines and functions should be written, or equivalents documented, as part of a migration strategy.